In the Specification:

Page 4 Lines 4 to 18

Figure 5 is a flow diagram of the steps carried out by an agent when evaluating a proposal;

Figure [[5]] 6 is a schematic diagram of calls for proposal from a buyer agent to seller agents;

Figure [[6]] 7 is a schematic diagram of unmoderated reverse auction between agents for buyers and sellers;

Figure [[7]] 8 is a schematic diagram of moderated reverse auction, between buyers' and sellers' agents and an intermediary;

Figure [[8]] 9 is a schematic diagram showing the architecture of a software agent for flexible negotiation;

Figure [[9]] 10 is a collection of four graphs illustrating the iterative negotiation process, based on different commercial issues, leading to a contract for purchase of a vehicle by a buyer from a seller;

Figure [[10]] 11 is a diagram of a FIPA compliant agent platform;

Figure [[11]] 12 is a diagram of an abstract agent shell, on the left-hand side, and an example of an agent, on the right-hand side, completed around that shell; and

Figure [[12]] 13 represents a FIPA standard iterated contract net protocol.

Page 22 Lines 3-24

The evaluation reasoner becomes active when an agent receives a proposal or counter-proposal from another agent. Upon receipt of such a message, the agent computes the utility it attains for the proposal. It uses an additive scoring function over each slot in the SLA where each slot is assigned a weight representing the relative

importance of that issue to that agent. For example, as illustrated in Figure 5, when an agent receives a SLA (Step 510) it goes through each slot in the proposal and computes a measure of desirability (a utility rating between 0 and 1) to the value contained therein (Step 512). The raw utility values are then multiplied by a weighting factor, [[(]]which indicates their relative importance[[)]], (Step 514) and then summed over all the slots (Step 516). This process produces a single utility value for the proposed SLA. In parallel, the agent sends the offer just received to the tactical reasoner (Step 518) to see what offer the agent would produce next using its current strategies and tactics. Once computed, this offer is returned to the evaluation reasoner and rated using the aforementioned scoring function (Steps 520 to 524). The computed utility values of the proposal and offer are compared (Step 526) and, [[If]]if the utility of the offer the agent would have sent is less than the utility of the offer just received, the offer is accepted (Step 528). Acceptance involves a conditional commitment by the server that it will execute the specified service under the SLA's terms and conditions. The commitment is conditional in that the client still has to confirm or deny the contract. Assuming the client confirms the contract, it then terminates all other negotiation threads for the same service instance. The second outcome of the SLAs evaluation is that the proposal is rejected. This occurs when: (i) the deadline for reaching an agreement has been reached; or (ii) another agent has been selected to perform the service. The final evaluation outcome is that the offer is neither accepted nor rejected. In this case, the agent generates a counter offer.

Page 33 Lines 11 to 13

An embodiment of the present invention will now be described with reference to Figures ~~5 to 9~~ 6 to 10, but with reference to the agent enabling technology of our GB-A-2332288, and to the description above of flexible agent based negotiators.

Page 38 Line 25 to Page 39 Line 3

The functional architecture for a FIPA-compliant agent platform is illustrated in Figure [[10]]11. The core facilities of FIPA 97 are supported, enabling semantic interoperability between agents on platforms from different manufacturers. Points of

interest regarding existing FIPA-compliant platforms for constructing the Reverse Auction application include:

Page 39 Lines 13 to 23

An agent "shell" is illustrated in Figure [[11]]12, which allows agents to be built from a small set of components. The AMS, DF and ACC illustrated in Figure [[10]] 11 are all constructed by adding components and domain specific information to the agent "shell". The lower three components are fundamental to FIPA compliant agents, and these deal with the messaging construction/parsing. The upper three components categorise the agent, its domain, its goals, and any special abilities it might have (e.g. negotiation, mediation, brokerage, etc.). This agent "shell" will be used to compose the buyer, seller, intermediary and brokerage agents described in this patent application. The example agent in Figure [[8]]9 shows the core components used to construct a buyer, seller or intermediary agent. The Negotiation Engine refers to the component with the ability to generate bids and counter bids.

Page 39 line 24 to Page 40 line 2

A software agent is illustrated in Figure [[8]] 9, and its architecture will now be described, from a static high level view of the components required to implement the buyer, seller or an intermediary agent. These components complement the components which are provided by the agent infrastructure technology that supports, amongst other services, the ability to communicate with other agents via speech-act based (XML encoded) asynchronous messages.

Page 40 lines 3 to 5

The core functional components of the reverse auction agent architecture are the Transaction Engine and the Negotiation Engine. These components can be visualised as special skills in the agent "shell" instance shown in Figure [[11]] 12.

Page 41 Lines 9 to 16

Negotiation Results includes an audit trail of the bids sent/received and counter-bids sent/received and the final result of each negotiation. These negotiation results can be used to for analysis on the performance of an agent such that its portfolio of negotiation profiles can be "tuned" for more effective negotiation in the future. The process of visualising and analysing the negotiation results is illustrated as the Neg Profile Optimiser on the architecture diagram. The results of the analysis can be used to directly update the Negotiation Profiles via an XML editor as illustrated in Figure [[8]] 9 as described in the later section Negotiation Profiles.

Page 42 Lines 12 to 19

The Transaction GUI forms the interface point between the agent whether it represents the buyer, seller or intermediary and provides service specific GUI to enable the user to specify what type of product they wish to buy or sell. This GUI forms the trigger the service Use Case scenarios. The GUI will enable the human user to encode the task it wishes to delegate to its agent. For example, such tasks could be to buy a Japanese sports car for under 10K pounds (described below with reference to Figure [[9]]10) or buy a Shania Twain CD for the least amount possible although it must be delivered within 2 days.

Page 48 Lines 10 to 20

The agent platform illustrated in Figure [[10]]11 includes an agent (DF - Directory Facilitator) which performs a yellow pages service for software agents. This service can be used by other agents to advertise the services that they offer such that other agents can query the knowledge of the DF agent to establish which agents it needs to interact with to perform a given task. Further, it is possible to utilise brokerage functions with the DF such that agents could register interest in agents providing particular services (e.g. agents selling cars) so that when a agent publishes information that meets the requirements of a given agent, all the registered agents are informed of the introduction

of the new service offered and given a identifier for the agent offering the service, such that it can be contacted to initiate the Reverse Auction process.

Page 50 Line 25 to Page 51 Line 12

The Interaction Use Cases discussed below represent the groups of agents that will interact with each other. The communications layer of the agent platform supports the exchange of messages between the agents in the forms illustrated in the Use Cases. The means to construct and understand the messages exchanged is a feature provided by a component in the agent "shell" as illustrated in Figure [[11]]12. FIPA defines a number of standard interaction protocols that include a set of protocols for sequencing performatives for negotiation and auction domains. These standard protocols provide a co-ordination framework to help structure the interaction between the agents. These standard protocols help enable agents to determine what messages they are expected and also includes guidance on the types of messages to exchange when errors occur. For example, a negotiation iteration protocol known as Iterated-Contract-Net is illustrated in Figure [[12]]13 in AUML (Agent Unified Modelling Language). The terms used to label the arrows connecting the Initiator and Participant refer to the ACL performative (or message type in effect).

Page 54 Line 28

Illustrated in Figure [[5]] 6

Page 54 Line 33

Illustrated in Figure [[6]] 7

Page 55 Line 7

Illustrated in Figure [[7]] 8

Page 57 Lines 13 to Page 58 Line 2

"eDeals on Wheels" is an example, illustrated by way of its graphical output in Figure [[9]]10. This uses the negotiation profile given above. Each of the four graphs is displayed on screen, and represents, for each of four different commercial issues, the progress of the negotiation between buyer and seller, as a graph of the number of iterations against the relevant quantity for that issue. The Figure also indicates the total number of issues involved: the year the car was made (top left graph); the total cost of the car (top right); the delivery time for the car (bottom left); the vehicle grade (bottom right); the make and model of car (not shown), the colour of the car (not shown) and the feature list, such as accessories provided in the car (not shown). Once the graphs for buyer and seller have converged, agreement has been reached. For example, on the total cost issue, the buyer originally bid for a cost of 3,000, but allowed the bid to rise slowly and then more rapidly, towards the end of the negotiation process, to a maximum of 10,000. At the same time, the seller began at 13,000 but decreased very slowly, and then more rapidly at the end, to converge on the final price of 10,000.